

AD/

ESD ADDRESS

DRI Call No. ~~85679~~ 85679Copy No. 1 of 2 cys.

## Technical Note

1976-37

Microprocessor Realization  
of a  
Linear Predictive Vocoder

E. M. Hofstetter

J. Tierney

O. C. Wheeler

30 September 1976

Prepared for the Defense Communications Agency  
under Electronic Systems Division Contract F19628-76-C-0002 by

Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LEXINGTON, MASSACHUSETTS



Approved for public release; distribution unlimited.

ADA033686

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, for the Military Satellite Office of the Defense Communications Agency under Air Force Contract F19628-76-C-0002.

This report may be reproduced to satisfy needs of U. S. Government agencies.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

A handwritten signature in cursive script, reading "Raymond L. Loiselle".

Raymond L. Loiselle, Lt. Col., USAF  
Chief, ESD Lincoln Laboratory Project Office

MASSACHUSETTS INSTITUTE OF TECHNOLOGY  
LINCOLN LABORATORY

MICROPROCESSOR REALIZATION  
OF A LINEAR PREDICTIVE VOCODER

*E. M. HOFSTETTER*  
*J. TIERNEY*  
*O. C. WHEELER*

*Group 24*

TECHNICAL NOTE 1976-37

30 SEPTEMBER 1976

Approved for public release; distribution unlimited.

LEXINGTON

MASSACHUSETTS



## ABSTRACT

A microprocessor realization for a linear predictive vocoder is presented. The goal was a low power, low cost, compact special purpose realization of a narrow band speech terminal. The resultant design is a general purpose two bus structure running at a 150 ns cycle time using as the basic signal processing element four of the AMD 2901 CPE chips. This basic structure is augmented by a four cycle multiplier to allow for sufficient signal processing power. The design concessions that mark the LPCM as a special purpose machine designed to be a speech terminal are: limited I/O, and limited memory. The present design requires 162 dual-in-line packages, dissipates less than 45 watts and occupies about 1/3 cubic foot.



## TABLE OF CONTENTS

ABSTRACT	iii
I. INTRODUCTION - The Design of a Microprocessor Based LPC Vocoder	1
II. LPCM SYSTEM DESCRIPTION	4
2.1 Architecture	4
2.2 Instruction Format	8
2.3 Data Memory Addressing	11
2.4 Timing Considerations	12
III. ENGINEERING CONSIDERATIONS	12
IV. DEBUGGING AND TEST SYSTEM	13
4.1 Hardware and Software Debugging Aids	13
4.2 The LPCM Simulator and Assembler	16
V. FIRMWARE CONSIDERATIONS	16
5.1 The LPC Algorithm	16
5.2 Implementation of the LPC Algorithm	19
VI. SUMMARY AND CONCLUSIONS	22
APPENDIX A: LPCM Mnemonics	25
APPENDIX B: LPCM Specifications	29
ACKNOWLEDGEMENTS	32
REFERENCES	33

## I. INTRODUCTION - The Design of a Microprocessor Based LPC Vocoder

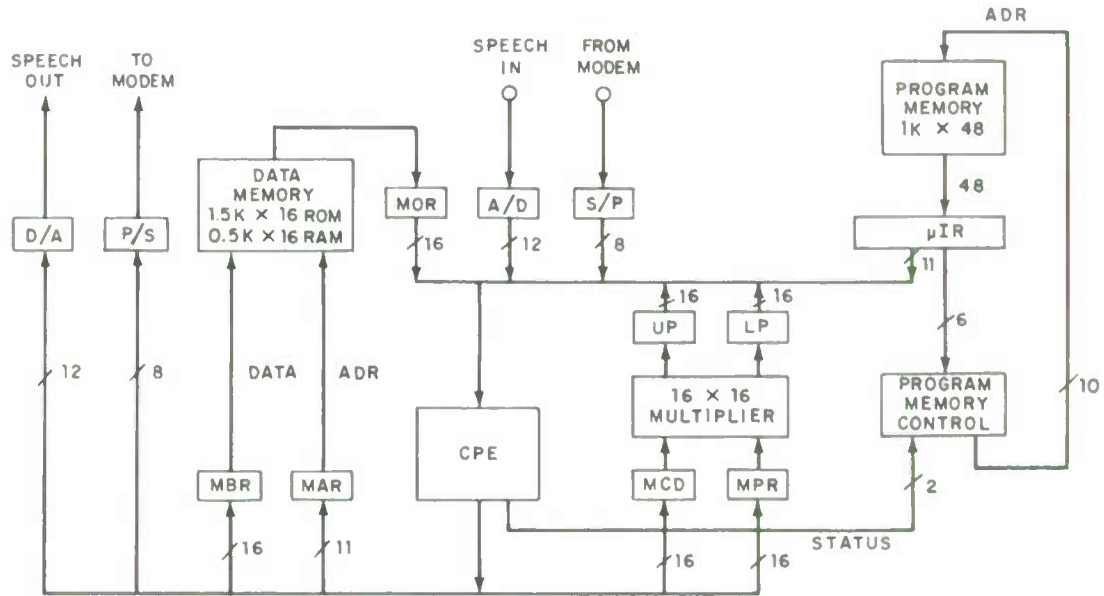
For the past several years there has been a trend toward the realization of narrow band speech terminals in the form of small general purpose digital computers. These computers have been fast enough to run the "real time code" necessary to transform them from general purpose computers to speech terminals capable of full duplex operation between talker-listener and modem. This approach was necessitated by the flux in narrow band speech algorithms during this time. As a result of recent work in linear predictive coding (LPC) techniques<sup>1,2</sup> applied to the analysis-synthesis of speech it has become possible to specify an LPC approach<sup>3</sup> which produces acceptable narrow band speech in the range from 2.4 to 4.8 Kb/s. In addition, a recent project at Lincoln Laboratory<sup>4</sup> provided the opportunity to implement the pertinent LPC code, pitch detector code, and data handling code in a very "lean" manner in terms of program and data memory use, and efficient real-time operation. This previous experience has enabled us to approach the design of a microprocessor based LPC vocoder with full knowledge of each subroutine and all timing sequences needed for interaction with both the incoming and outgoing audio data as well as the outgoing and incoming digital data stream.

Our starting goals for a microprocessor realized linear predictive vocoder were the production of a compact, low power, inexpensive device using commercially available integrated circuits. We were willing to design a completely special purpose device<sup>5</sup> that would implement only the LPC voice terminal in an efficient form. In addition there was no consideration of custom large-scale-integration chip use since the costs for a limited vocoder market



appeared too high, and no small set of chip types seemed adequate. In effect the goal was a benchmark device using only commercial chips whose price would drop with the larger commercial market. This benchmark device could then be used in larger system designs as a cheap building block, or could be modified and expanded to include modem and other functions.

Starting with a study of available microprocessor chip sets a particular choice was made on the basis of speed, signal processing power, and basic chip organization (the AMD 2900 series). Several design iterations were then made starting with a machine using three separate microprocessor CPE's. In this design each CPE was doing a special purpose task, and was fed from separate analog processing circuits. Because of inefficiencies associated with memory sharing and access, this design evolved to a two CPE machine with the machine physically divided into a transmitter and separate receiver. This design also appeared inefficient. Finally it was seen that a single CPE and hardware multiplier could satisfy all of the signal processing requirements for the given algorithms. A complete software study then preceded the detailed logic design. In effect all of the machine code was written or blocked out to verify the design. In spite of our avowed goal of a special purpose vocoder device we have in the end designed a rather general purpose structure. The limited in-out capability as well as the limited data and program memory are what remain of the special purpose device. The end design is based on a single microprocessor CPE augmented with a four cycle multiplier. The basic structure is that of a two bus G.P. machine with separate program and data memory as shown in Figure 1.



LPCM BLOCK DIAGRAM

Fig. 1. LPCM block diagram.

## II. LPCM SYSTEM DESCRIPTION

### 2.1 Architecture

The basic block diagram for the LPCM is shown in Figure 1. All instructions for this machine are executed in a 150 ns cycle except the multiply which requires four machine cycles or 600 ns. The nucleus of this system is the CPE which is based on the AMD 2901 microprocessor chip. Four such chips are used along with a carry-lookahead chip to yield a 16-bit CPE.

A simplified block diagram of the 2901 appears in Figure 2. From this diagram it can be seen that the chip consists of an ALU capable of add, subtract and Boolean operations coupled with an internal 2-port general register file consisting of 16 words. Multiplexers at the input of this register file permit a 1-bit up or down shift prior to writing the memory. A Q-register is provided which allows double precision shifts to be implemented. Inputs to the chip from the outside world consist of two 4-bit addresses for the internal register file, control signals and data from external devices such as memory or I/O devices. The manufacturers' literature should be consulted for further details about the 2901.

Referring again to Figure 1, it is seen that the 16-bit CPE is connected to an input and an output data line. The input line is multiplexed between 6 data sources, the 16-bit memory output register (MOR) of the data memory, the 12-bit A/D converter, the 8-bit serial-to-parallel (S/P) converter, the 16-bit upper and lower products coming from the multiplier and an 11-bit field coming from the instruction register. The data memory consists of 2K 16-bit words 1.5K of which are ROM and contain the various lookup tables needed to implement the LPC algorithm. The output of the CPE is channeled to the D/A converter,

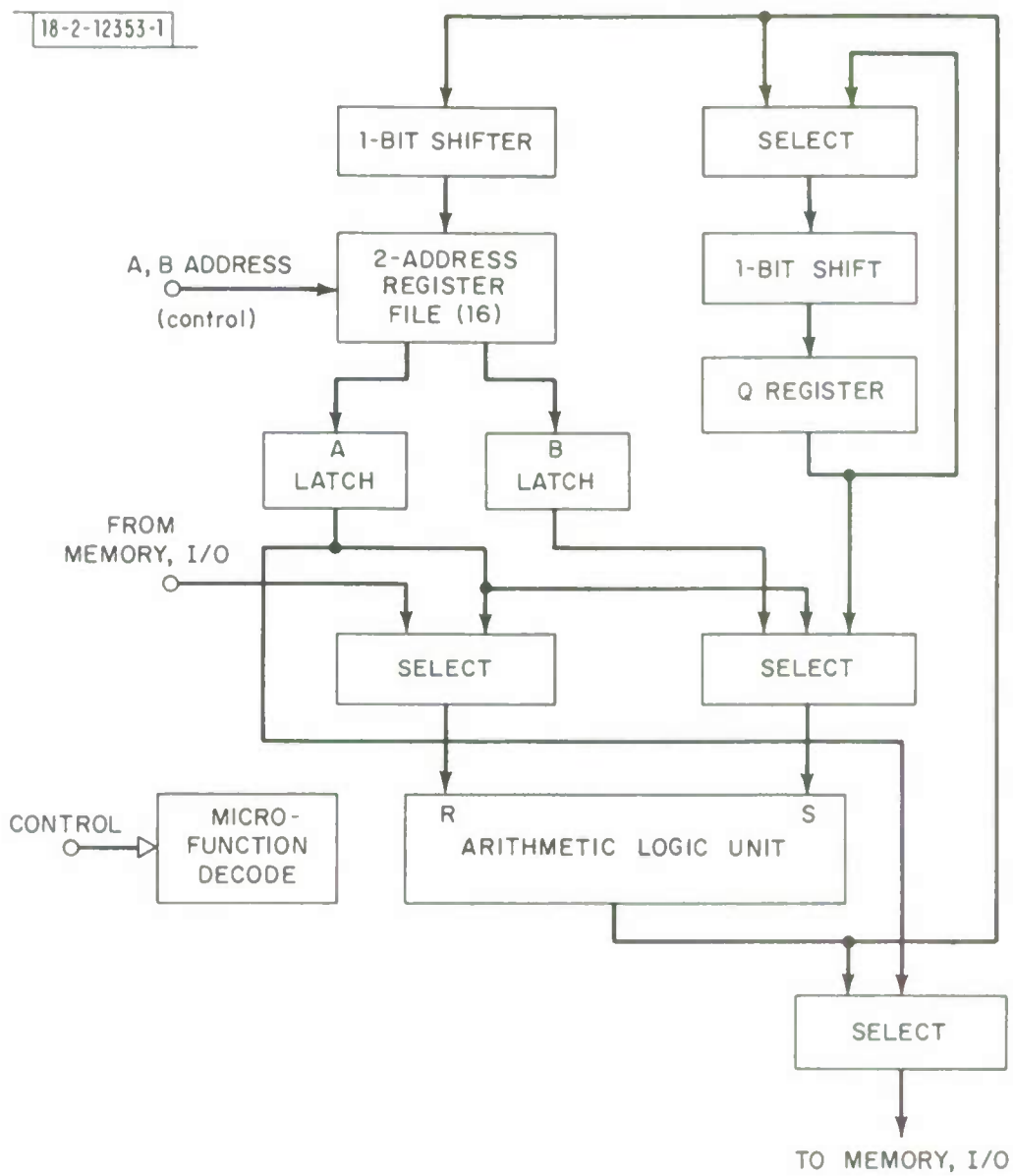


Fig. 2. CPE chip block diagram.

the parallel-to-serial (P/S) converter, the memory buffer and address registers (MBR and MAR) and the multiplicand (MCD) and multiplier (MPR) registers of the multiplier. These various output registers are clocked under the control of a 3-bit field in the instruction register.

The multiplier uses the Booth-McSorley algorithm to multiply two 16-bit two's complement numbers and makes the full 32-bit product available to the CPE's input ports in two 16-bit pieces. The multiplier is fabricated from the AMD25S05 4x2 multiplier chip. Eight of these are used to construct a 16x4 array multiplier which is clocked four times to yield the final product. The outputs are fully buffered so that the product may be retrieved from the multiplier any time four machine cycles or longer after the start of the multiply. The CPE is free to do other tasks in this interval while multiplication is taking place.

The program memory contains 1K of 48-bit words. The output of this memory is clocked into a microinstruction register and the memory address is derived from the program control logic. The latter is based on the AMD2909 program sequencer chip, a simplified block diagram of which appears in Figure 3. Three of these 4-bit chips are used making it possible to address 4K of program memory even though only 1K of such memory is needed for the present application. The 2909 controller is driven by a 2-bit control line which enables one to select the next program address to be either the last address plus one, a jump address which comes from the microinstruction register, the latest address on the internal stack, or an interrupt address determined by the I/O system. The jump logic which drives the control ports of the 2909 allows for unconditional

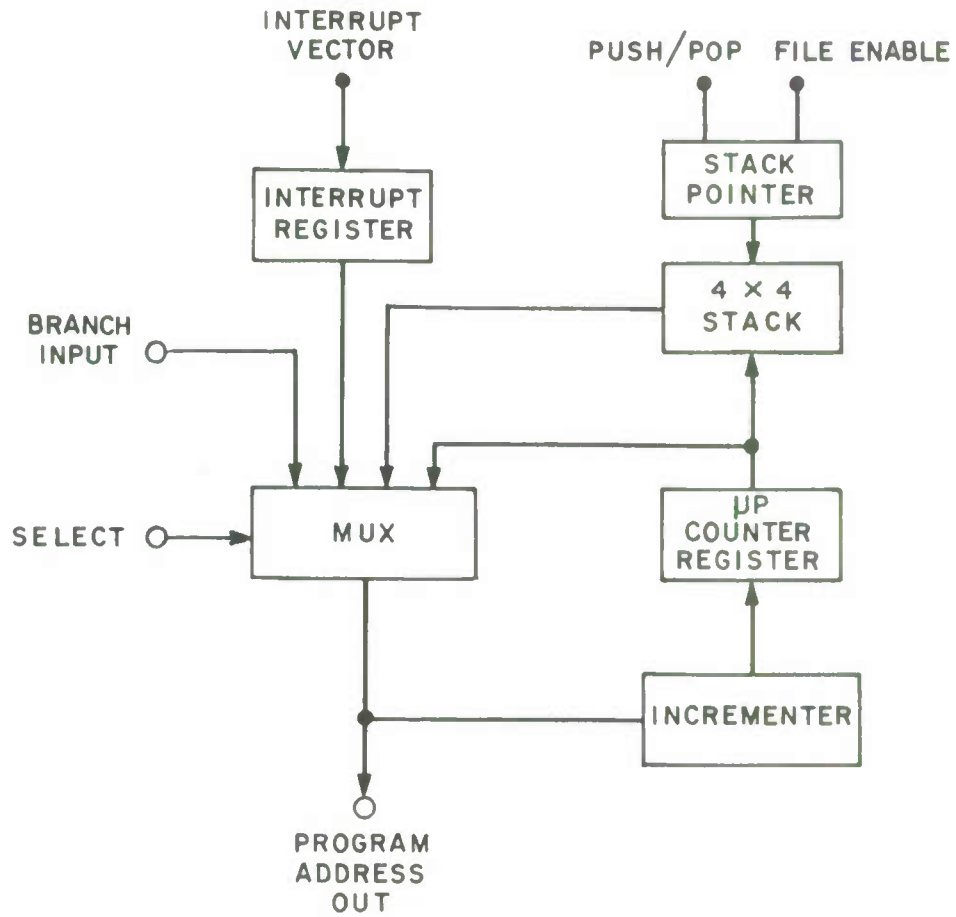


Fig. 3. Program sequencer chip block diagram.

jumps, conditional jumps depending on the status bits coming from the CPE and jumps to and returns from subroutines. Subroutines may be nested up to four deep when interrupts are locked out and three deep when they are active.

The I/O system for the LPCM consists of two input channels, the A/D and S/P converters, and two output channels, the D/A and P/S converters. The A/D-D/A channels run on a common 129.6  $\mu$ s clock that is derived from the 150 ns system clock. The P/S and S/P converters run on external modem clocks which must have the same nominal frequency (2400, 3600 or 4800 Hz) but which may be asynchronous to one another. The I/O channels generate an interrupt request whenever their associated clocks present a rising edge to the system. This request causes the program control logic to produce a jump to one of three predetermined locations in program memory at the first instance the system finds itself in a position to allow interrupts. Several interrupts may have requests pending at one time; they are serviced in order of their priorities which are P/S, S/P and A/D-D/A. While a given interrupt is being serviced, all others are locked out. Upon return from an interrupt service routine the software releases interrupt lockout thus enabling the honoring of further interrupt requests.

## 2.2 Instruction Format

The format of the 48-bit wide instruction word is shown in Figure 4. The instruction word is divided into various fields of varying length the functions of which will now be discussed.

The  $C_O$ ,  $I_S$  and  $I_O$  fields determine the basic operation the CPE is to perform, e.g., add the contents of internal register at address A to the contents

18-2-13099

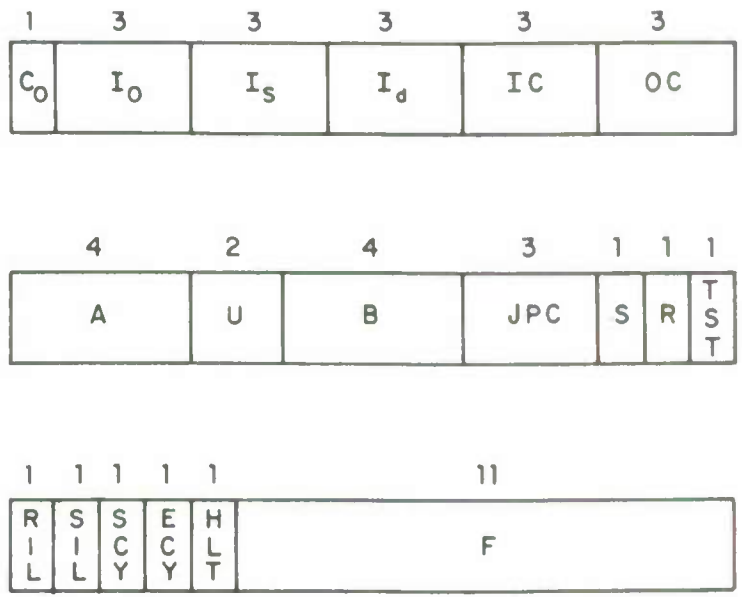


Fig. 4. LPCM microinstruction word format.



of the internal register at address B or take the external data presented to the chip and logically AND it with the contents of the internal register at address A. A list of useful combinations of these fields along with a mnemonic for each is given in Appendix A.

The  $I_d$  field determines where on the CPE chip the output of the ALU is to go. Some examples are: the output of the CPE alone, the output of the CPE and internal register file at address B, or the output of the CPE and the Q register.

The IC and OC fields determine where the CPE gets its input and where its output is to go, respectively. The IC field steers the input 6-way multiplexer to any of the input sources mentioned above and the OC field determines which, if any, of the output registers connected to the CPE are to be clocked. The A and B fields simply supply the addresses to the CPE's two-port memory and need no further discussion.

The JPC field along with the R and S fields provides program control by means of various kinds of jumps. A complete list of these appears in Appendix A. Conditional jumps in the LPCM are somewhat unconventional in that the condition on which the jump is to be based must be established in an instruction preceding the actual jump instruction by means of the TST field. More precisely, if one wishes to conditionally jump, say, based on whether one of the CPE's internal registers is zero, then the contents of this register must be made to appear at the CPE output with an instruction that also has the TST bit set. This strobes the CPE status into a (2-bit) status register which in turn may be tested by a subsequent instruction containing the appropriate jump code.

The remaining fields are quite straightforward. The F field appears directly at the CPE input where it can be used for a constant or a base address. This field also contains the jump address and must be set accordingly for each instruction containing a jump. The SIL and RIL fields are used to set interrupt lockout and release interrupt lockout, respectively, and are primarily used to prevent interrupts while executing calculations that an interrupt could destroy such as an ongoing multiply. The SCY and ECY fields are provided to facilitate multiple-precision adds and subtracts. When the SCY bit is set during an add or subtract instruction, the carry resulting from this operation is saved in a flip-flop. This saved carry can then be used in a later add or subtract instruction by setting the ECY bit during that instruction. Finally, the HLT bit stops the machine; a feature that is only used during debugging operations. The two bits labelled U are unused.

### 2.3 Data Memory Addressing

Addresses for the LPCM data memory must be generated in the CPE and then deposited in the MAR. Direct addressing of data memory is achieved by having the desired address in the F field of the microinstruction word and passing it through the CPE to the MAR. Indexed addressing can be accomplished by having a base address in the F field, adding to it the contents of a CPE internal register and depositing the result in the MAR. It should be noted, however, that the contents of the addressed location in data memory are only available as a CPE input one instruction cycle after the desired address is placed in the MAR. This is due to the fact that the memory output is buffered in the MOR. Writing data memory is also a 2-step process in the sense that the

address must first be calculated and deposited in MAR before the datum itself may be read out into the MBR.

#### 2.4 Timing Considerations

The basic events that must take place in order to execute an LPCM instruction are as follows:

- a) program counter assumes desired state
- b) program memory is accessed
- c) accessed instruction is executed by CPE

It is not possible to perform all three of these operations in the desired cycle time of 150 ns so the sequence is broken into two parts by inserting the microprogram instruction register after the program memory. This results in what is called a doubly-overlapped pipeline structure in which instruction fetch takes place in parallel with execution of the instruction fetched on the previous machine cycle. This type of pipelining is transparent to the programmer of the LPCM.

The LPCM also employs pipelining in the data memory acquisition path and in the jump control path as has been described earlier. This pipelining is not transparent to the programmer in that memory addresses and jump conditions must be set up sufficiently in advance of the instruction that makes use of them. Experience has shown that careful programming can usually circumvent any potential loss of program efficiency caused by these pipelined paths in the machine.

### III. ENGINEERING CONSIDERATIONS

The present LPCM is a prototype designed to demonstrate that

a dedicated linear predictive vocoder can be realized both cheaply and compactly using off-the-shelf components. Since it is a prototype it was decided to use standard 16x7 inch universal wirewrap boards as the packaging medium rather than go directly to smaller PC boards. Universal boards were chosen because the LPCM uses every standard package size from 14-pin to 40-pin in its design. The final design uses 162 DIPS and occupies 1.5 boards. These figures include all of the analog circuits required before and after the A/D and D/A converters. The power consumption of the device is less than 45 watts. A photograph of the completed LPCM appears in Figure 5.

Appendix B gives a complete compilation of the parts used to fabricate the LPCM. Included in the table are military and commercial cost figures for building 1, 500, 1000 and 10,000 processors. These figures are based on the extrapolation rules provided by the Narrow Band Voice Consortium Subcommittee for estimation of "cost to produce". The figures referring to the packaging of the LPCM are estimates of how it could be packaged using PC boards and do not reflect the present wirewrap packaging of the prototype.

#### IV. DEBUGGING AND TEST SYSTEM

##### 4.1 Hardware and Software Debugging Aids

The LPCM is intended to be a stand-alone device with its control program residing in PROM's. During the debugging phase, however, it is necessary to replace the PROM memory with RAM in order to facilitate program changes and allow the running of diagnostic programs. In addition, it is extremely advantageous to have a means for starting and stopping the machine, setting breakpoints and examining the contents of data memory and the CPE's internal register file.

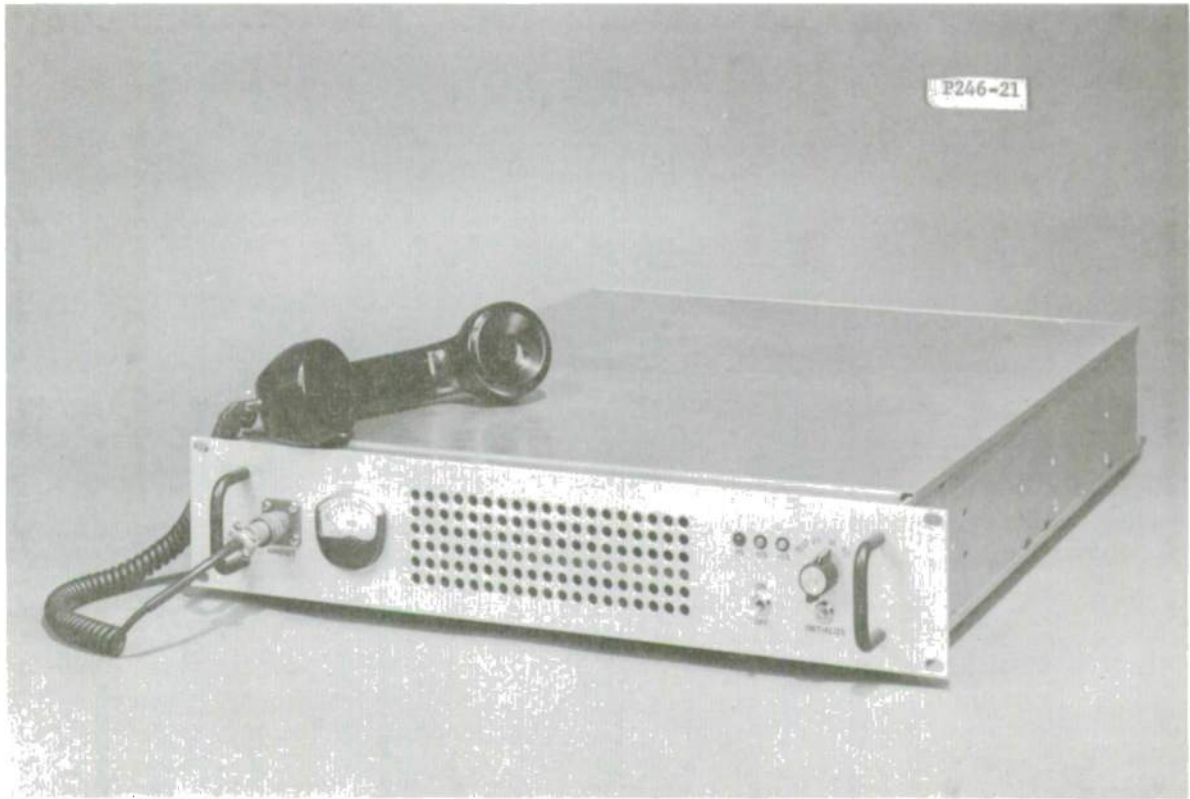


Fig. 5. The completed LPCM.

The above requirements were met by the design and fabrication of a separate unit - the LPCM tester - which is connected to the LPCM by means of cables during the debugging phase. The main component of the tester is a 1024x48 RAM which effectively replaces the PROM memory destined to reside in the LPCM. In addition, the tester duplicates the AM2909 program control chips that are located in the LPCM itself. This was done to minimize both the number of control cables between the LPCM and its tester and the tester-oriented logic needed in the LPCM.

The tester's program memory can be loaded in either of two ways; a) one register at a time by means of front-panel switches or b) the entire memory can be loaded from a host computer. The first mode is useful for toggling in small test programs and patching larger programs. The latter mode is used for loading large programs such as the diagnostic system or the LPC vocoder program itself. When the tester is connected to the LPCM the following control functions are available.

- a. start program at an arbitrary address
- b. stop program
- c. single-step program
- d. stop at breakpoint determined by switches
- e. inspect any location in data memory
- f. inspect any location in CPE register file
- g. inspect/change any location in program memory

In addition to the above mentioned hardware debugging aids, an extensive software diagnostic system was written for the LPCM. This system tests the following functions of the LPCM:

- a. RAM portion of data memory
- b. CPE functions
- c. Jump logic
- d. Multiplier
- e. I/O



#### 4.2 The LPCM Simulator and Assembler

A simulator for the LPCM was written on a Univac 1219 computer so that software debugging could take place in parallel with the fabrication of the LPCM hardware. The simulator accepts as its input the binary code generated by an LPCM assembler. This assembler was also written on the Univac 1219 and is a straightforward two-pass assembler that understands LPCM mnemonics and symbolic addresses. Symbolic code is generated using the Univac's editor and then fed to the assembler which produces a binary output that can be loaded into the LPCM or operated on by the simulator. This same binary output was later used to burn in the PROMs that comprise the LPCM's program memory.

The simulator is fairly sophisticated in that it simulates all I/O operations including interrupts. This allowed the debugging of not only the diagnostic package but the entire LPC vocoder program itself. In the final stages of the vocoder programming, real speech was used as the input to the simulator and the synthetic speech output of the program was stored on magnetic tape. All computation was done in non-real time but the final output tape was then played back in real time to provide convincing evidence that the LPCM vocoder algorithm was functioning correctly. This indeed proved to be the case for, when the program was finally running on the LPCM itself, only a few additional program bugs were found.

### V. FIRMWARE CONSIDERATIONS

#### 5.1 The LPC Algorithm

LPC was first described by Atal and Hanauer in 1971<sup>1</sup>. Since then many variations on this algorithm have appeared in the literature (see

bibliography in (2) and (6)). We have chosen to implement the Markel form of the LPC algorithm for reasons detailed in (7).

This algorithm is described in block-diagram form in Figure 6. Speech samples taken every 129.6  $\mu$ s are divided into 158-point non-overlapping groups corresponding to approximately 20 ms of data. These groups are multiplied by a Hamming window and then used to form  $P+1$  autocorrelation coefficients  $R_0, \dots, R_P$ . The parameter  $P$  is the order of the filter used to model the vocal tract and ranges from 10 at 2400 BPS to 12 at 3600 and 4800 BPS.

The autocorrelation coefficients are used as the constants in a set of linear equations that must be solved to obtain the parameters of the vocal tract filter. These equations are solved by means of the Levinson recursion<sup>8</sup> which yields a set of  $P$  reflection coefficients  $K_0, \dots, K_{P-1}$  and a residual energy  $E$ . These reflection coefficients will be used at the receiver to implement the vocal tract filter. The structure chosen for this filter is the acoustic tube filter described in detail in (2). The residual energy is used at the receiver to generate the amplitude of the excitation for the acoustic tube.

In addition to the processing described above, the raw speech samples are fed to a pitch and voicing detector which produces both a voiced-unvoiced decision and an estimate of pitch. The particular algorithm used for this purpose is the Gold-Rabiner pitch detector which is described in detail in (9) and (10).

The parameters produced as described above are next coded and formed into a serial bit stream for transmission to the remote receiver. The receiver



18-2-12578-1

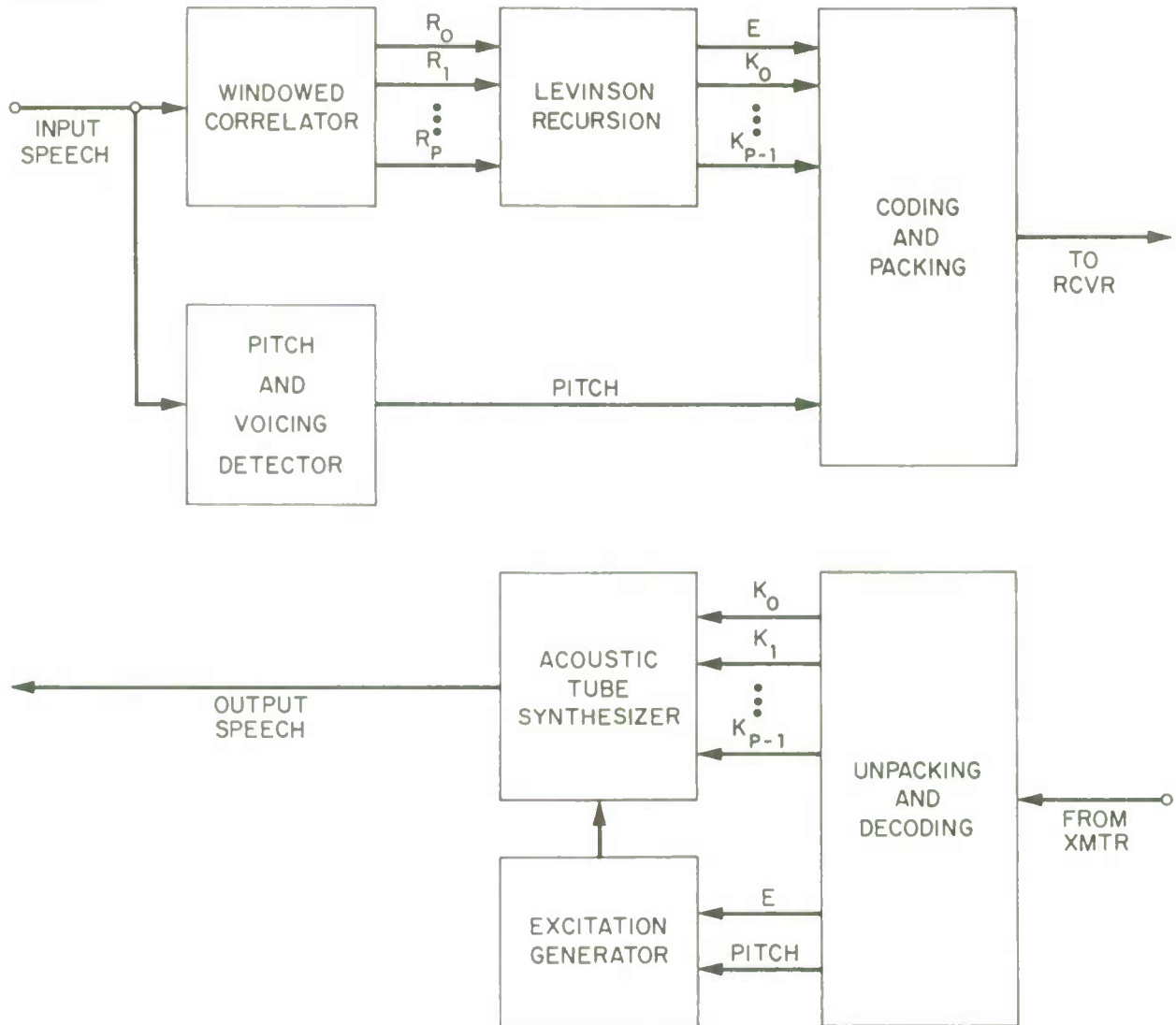


Fig. 6. The LPC vocoder algorithm.

portion of the algorithm accepts such a serial bit stream from the remote transmitter and unpacks it to form the code book addresses of the various parameters. These addresses are then decoded to obtain the actual values of the parameters which are then used to implement the acoustic tube filter and its excitation. The output of the filter is the final synthetic speech.

The coding of the parameters, except for pitch which is transmitted as is, is accomplished by a logarithmic-search table-look-up routine. The residual energy is logarithmically coded to 5 bits. The reflection coefficients are coded by means of truncated, log-area ratios. Each reflection coefficient is first clamped to an individually selected interval, transformed by the log-area-ratio function ( $\log [(1-K)/(1+K)]$ ), and finally truncated to the desired number of bits. The number of bits used for the individual K's is a function of the desired transmission rate.

## 5.2 Implementation of the LPC Algorithm

The LPC program consists of four major pieces, a background program that handles all of the computation that need only be performed once per frame and three interrupt service routines that handle the computations that must be done for each modem clock and each A-D/D-A clock.

The A-D/D-A interrupt service routine uses the newly arrived speech sample to update the current windowed correlation and the six elementary pitch detectors. In addition the acoustic tube filter is updated to produce a new synthetic speech sample for the D/A converter. This approach eliminates the need for any substantial buffering of raw speech thus reducing our data memory requirements. The reflection coefficients for the acoustic tube are interpolated against the coefficients for the next frame every 5 ms and the

amplitude is interpolated every time a new pitch pulse is generated. No amplitude interpolation takes place during unvoiced frames.

The main task of the P/S converter interrupt service routine is to pass the coded data produced by the analyzer portion of the program to the transmit modem. This is accomplished by loading the first code word into the P/S converter and then counting a number of interrupts equal to the known number of bits in this word. Subsequent words are then loaded and the appropriate number of interrupts counted after each. When a complete frame of code words has been serialized in this fashion and passed to the transmit modem, the current correlation coefficients are transferred to registers used by the background routine, the correlator is reset to start a new correlation and a flag is set to tell the background routine to start a new frame calculation using the new correlation coefficients.

The S/P interrupt service routine receives serial data from the receiver modem. It deserializes this stream into the proper length code words using an interrupt counting technique similar to the one used by the P/S converter. The code words are then used to access decoding tables thus producing the parameters eventually used by the acoustic tube synthesizer. These parameters are transferred to the buffer used by the acoustic tube when the S/P routine's counters determine that it has received a complete frame of new data.

The deserialization procedure just described only makes sense if the S/P routine "knows" where the first code word of a frame is in the incoming bit stream. The process of making this determination is known as

frame synchronization and is another task of the S/P routine. Frame synchronization is established by having the transmitter transmit a known bit pattern in place of the pitch word during unvoiced utterances. The pattern is chosen to correspond to an illegal (too high) pitch so that the receiver can still make an unambiguous buzz/hiss decision. The frame synchronization algorithm now consists simply of searching for this known pattern in the serial bit stream as it arrives at the receiver. Synchronization is declared (i.e., knowledge of the location of the pitch word) when, and only when, the known pattern has been found at the same location in six consecutive frames. When this occurs, the S/P routine sets its bit and word counters accordingly thus establishing synchronization.

The final routine to be discussed is the background routine. The start of this routine is an idle loop whose sole purpose is to continually check the status of the frame ready flag that is set by the P/S interrupt service routine. As long as this flag is clear, the program remains in the idle loop except for those times when an interrupt arrives and transfers control to the appropriate service routine. When the flag is finally set, the program drops out of the idle loop and begins its once-a-frame computations. The first of these is the final determination of pitch by a routine that examines the status of the six elementary pitch detectors and produces a buzz/hiss decision and an appropriate pitch. Next, the double-precision correlation coefficients are put into a block-floating point format based on  $R(0)$  and passed on to the Levinson recursion which produces the desired reflection coefficients and the residual energy. The latter is unnormalized to remove the scale factor introduced by the block floating-point routine and then the parameters are coded using

the appropriate coding tables. The final code words are placed in a buffer where the P/S routine can access them for shipment to the transmit modem. Control is then returned to the idle loop. It should be emphasized that while the background routine is calculating, interrupts are active which means that the background routine is only actually working in the intervals when no interrupt service routine is in progress.

One final routine should be mentioned and that is the initialization routine. This routine starts at program address zero and is only entered on power-up or when the initialize pushbutton is pressed. The main function of this routine is to clear data RAM, initialize the few RAM registers that require it and finally determine which rate vocoder is desired. The latter function is accomplished by sensing a front panel rate-control switch and then setting pointers to the proper coding and decoding tables. In addition, if the rate selected is 2400 BPS, the filter order is changed from 12 to 10.

## VI. SUMMARY AND CONCLUSIONS

We have presented the motivation and realization for a microprocessor based linear predictive vocoder. The resultant device is an existence statement for low power, low cost, compact digital realizations of narrow band speech terminals. What began as an exercise in the design of a special purpose digital machine for narrow band speech has ended with a general purpose two bus structure running at a 150 ns cycle time, using as the basic signal processing element four of the AMD 2901 four bit CPE microprocessor chips. This basic sixteen bit CPE is augmented by a four cycle hardware multiplier to allow for sufficient signal processing power. The design concessions that

mark the LPCM as a special purpose machine designed to be a speech terminal are: limited I/O capability, and limited data and program memory. The I/O bus only communicates with A/D-D/A, parallel-to-serial modem input and serial-to-parallel modem output. The LPCM data memory consists of 1536 locations of 16-bit ROM tables and 512 locations of 16-bit RAM words. The program memory consists of 1K by 48-bits of ROM of which less than 800 locations are used. A priori knowledge of the operating algorithms as well as an operating simulator and diagnostics reduced the entire time from design to completion to less than one year. The present package requires 162 DIP's including audio circuits, dissipates less than 45 watts, and occupies about 1/3 cubic foot. The operating code occupies the machine for about 65% of real time.

As a prototype device the LPCM specifications are not as tight as they might be. Given the 65% utilization, the cycle time can be slowed to over 200 ns and power dissipation reduced by roughly 10 watts. The volume can be reduced by as much as a factor of 3 if printed circuit boards are used, and tighter packaging is designed.

The overall package count of 162 various sized DIP's includes the seven packages of AMD CPE (4) and AMD sequencer (3), about 40 packages of memory and memory related circuits, 20 packages for multiplier, and the rest for I/O, bus multiplexing, timing, interrupt and branching. It is clear that in terms of power and size the device is not defined by the microprocessor chips. The overall machine size is determined by all of the "glue logic" and memory packages which swamp out the microprocessor chips. In fact the

memory and memory related packages probably represent a lower bound on size and power, in the sense that everything else may shrink considerably, but the current memory size and power are relatively static.

## APPENDIX A: LPCM Mnemonics

The following is a compilation of the bit assignments that must be made to the fields of the LPCM microinstruction word to achieve various functions. Each of these assignments is preceded with a mnemonic that can be used when preparing code for the LPCM assembler. The first group of these assignments are the so called "op codes" which affect the  $C_o$ ,  $I_o$  and  $I_s$  fields. The format of the presentation consists of a mnemonic followed by a three digit octal number giving the values assigned to  $C_o$ ,  $I_o$  and  $I_s$ , respectively, followed by a brief description of the operation accomplished by the assignment. The result of the operation appears at the internal ALU output port. The following notation is used in the descriptions.

R(A)	contents of internal register addressed by the A field.
R(B)	contents of internal register addressed by the B field.
Q	contents of the Q register.
D	data at input port of the CPE
.	logical and
!	logical or
@	logical exclusive or
%	logical complement

It should be noted that all possible operations that the CPE is capable of are not included in the following list.



ADDAB	001	$R(A) + R(B)$
ADDDA	005	$D + R(A)$
ADDAB 1	101	$R(A) + R(B) + 1$
ADDDA 1	105	$D + R(A) + 1$
SUBBA	111	$R(B) - R(A)$
SUBAD	115	$R(A) - D$
SUBAB	121	$R(A) - R(B)$
SUBDA	125	$D - R(A)$
SUBBA 1	011	$R(B) - R(A) - 1$
SUBAD 1	015	$R(A) - D - 1$
SUBAB 1	021	$R(A) - R(B) - 1$
SUBDA 1	025	$D - R(A) - 1$
MOVB	033	$R(B)$
MOVA	034	$R(A)$
MOVD	037	$D$
INCB	103	$R(B) + 1$
INCA	104	$R(A) + 1$
INCD	107	$D + 1$
DECB	013	$R(B) - 1$
DECA	014	$R(A) - 1$
DECD	027	$D - 1$
CSB	123	$-R(B)$
CSA	124	$-R(A)$
CSD	117	$-D$
ANDAB	041	$R(A) \cdot R(B)$
ANDDA	045	$D \cdot R(A)$
ORAB	031	$R(A) ! R(B)$
ORDA	035	$D ! R(A)$
XORAB	060	$R(A) \oplus R(B)$
YORDA	065	$D \oplus R(A)$
CMPB	023	$\%R(B)$
CMPA	024	$\%R(A)$
CMPD	017	$\%D$
CLR	142	$\emptyset$

The next set of assignments concerns the destination field,  $I_d$ , which determines where the output of the ALU is to go. The format is mnemonic, one digit octal number and description. The notations F for ALU output and Y for CPE output are used in the descriptions.

Q	0	$F \rightarrow Q, F \rightarrow Y$
Y	1	$F \rightarrow Y$
RAY	2	$F \rightarrow R(B), R(A) \rightarrow Y$
R	3	$F \rightarrow R(B), F \rightarrow Y$
SDD	4	double precision down shift $[F, Q]/2 \rightarrow [R(B), Q]$ $F \rightarrow Y$
SD	5	$F/2 \rightarrow R(B), F \rightarrow Y$
SUD	6	double precision up shift $[F, Q]*2 \rightarrow [R(B), Q]$ $F \rightarrow Y$
SU	7	$F*2 \rightarrow R(B), F \rightarrow Y$

The next set of assignments concerns the IC field which controls the input multiplexer to the CPE. The format is mnemonic, one digit octal number and description.

SP	0	serial-to-parallel converter
ADC	1	A/D converter
LP	2	bits 0-15 of the product
UP	3	bits 15-30 of the product
MOR	4	memory output register
FD	5	11 bit instruction field

The clocking of the various registers connected to the output of the CPE is controlled by the output control field OC. The format is the same

as for the input control field.

NIL	Ø	clock nothing
MAR	1	clock memory address register
MBR	2	clock memory buffer register
MCD	3	clock multiplicand register
DAC	4	clock D/A converter buffer register
PS	5	clock into P/S converter
MPR	6	clock multiplier register and start multiply sequence

The final group of assignments concerns the jump control fields, JPC, S and R. The format is mnemonic, three digit octal numbers giving the assignment to the JPC, S and R fields, respectively, and a description.

NIL	ØØØ	no jump
JP	100	unconditional jump
JPZ	200	jump if positive or zero
JZ	300	jump if zero
JN	400	jump if negative
JNZ	500	jump if not zero
JSW	600	jump if switch w on
JSV	700	jump if switch v on
JPS	110	unconditional jump to subroutine
JPZS	210	jump to subroutine if positive or zero
JZS	310	jump to subroutine if zero
JNZS	410	jump to subroutine if negative
JSWS	610	jump to subroutine if switch w set
JSVS	710	jump to subroutine if switch v is set
SBR	101	return from subroutine

## APPENDIX B: LPCM Specifications

<u>Cycle Time</u>	150 ns
<u>Basic Logic family</u>	TTL Using low power Schottky TTL
	in AMD chips, high power Schottky where necessary in critical paths.
<u>Program Memory (R.O.M.)</u>	1K x 48 bits      12 - MMI 6351 (1Kx4)
<u>Data Memory (R.O.M.)</u>	1536 x 16 bits    4 - MMI 6351 (1Kx4)
	2 - FCLD 93448 (512x8)
<u>Data Memory (active)</u>	512 x 16 bits      8 - FCLD 93442 (256x4)
<u>Hardware Multiplier</u>	One quarter of an array operating in 150 ns
4x16 multiply	8 - AMD 25S05 (2x4)
<u>Basic C.P.E.</u>	4 - AMD 2901 (4 bit slice)
<u>Microsequencer</u>	3 - AMD 2909 (4 bit slice)
<u>Audio Conditioning</u>	12 bit A/D, D/A conversion at 129.6 $\mu$ sec samples.
	Input Filter 8th order, elliptic filter 52 dB stop band attenuation
	1.2 dB ripple, cutoff at 3596 Hz.
	Output Filter 8th order, elliptic filter 41 dB stop band attenuation
	0.2 dB ripple, cutoff at 3596 Hz.
<u>Total DIP Count</u>	162
<u>Total Power Dissipation</u>	45 watts
<u>Construction Technique</u>	Two universal wire wrap boards (50% of 2nd board unused)
	7" x 16"
	center plane voltage
	two outside planes ground

ITEM	QUANT. PER UNIT	SOURCE	ITEM COST		1 PROCESSOR			500 PROCESSORS			1000 PROCESSORS			10,000 PROCESSORS		
			COMM.	MIL	QUANT. MULT.	COMM.	MIL	QUANT. MULT.	COMM.	MIL	QUANT. MULT.	COMM.	MIL	QUANT. MULT.	COMM.	MIL
7400	6	T1	0.86	3.27	6	5.16	19.62	3.96	3.18	12.12	2.82	2.43	9.22	2.10	1.80	6.86
7442	1	T1	3.05	9.16	1	3.05	9.16	.66	2.01	6.05	.66	3.71	6.05	.35	1.07	3.21
7408	2	T1	.98	3.53	2	1.96	7.06	1.32	1.29	4.66	.94	.92	3.32	.70	.69	2.47
7432	1	T1	1.73	4.65	1	1.73	4.65	.66	1.14	5.30	.66	1.14	5.30	.35	.61	1.63
7404	9	T1	1.41	4.04	9	12.69	36.36	4.23	5.96	17.09	4.23	5.96	17.09	3.15	4.44	12.73
7410	2	T1	1.14	3.27	2	2.28	6.54	1.32	1.50	4.32	.94	1.07	3.07	.70	.80	2.29
74S151	1	T1	3.04	22.11	1	3.04	22.11	.66	2.01	14.59	.66	2.01	14.59	.35	1.06	7.74
74S257	4	T1	3.97	21.24	4	15.88	84.96	1.88	7.46	39.93	1.88	7.46	39.93	1.40	5.56	29.74
74S00	2	T1	1.39	8.15	2	2.78	16.30	1.32	1.83	10.76	.94	1.31	7.66	.70	.97	5.70
74S02	6	T1	.88	8.15	6	5.28	48.90	3.96	3.24	30.24	2.82	2.48	22.98	2.10	1.84	17.12
74S04	5	T1	1.69	10.12	5	8.45	50.60	3.30	5.58	33.40	2.35	3.97	23.78	1.75	2.96	17.71
74S74	2	T1	2.79	14.55	2	5.58	29.10	1.32	3.68	19.21	.94	2.62	13.67	.70	1.95	10.19
74S174	25	T1	5.55	33.90	25	138.75	847.50	8.75	48.56	296.63	8.75	48.56	296.63	8.75	48.56	296.63
74S157	1	T1	3.75	21.24	1	3.75	21.24	.66	2.48	14.87	.66	2.48	14.87	.35	1.31	7.43
7414	1	T1	6.72	12.92	1	6.72	12.92	.66	4.16	7.99	.66	4.16	7.99	.35	2.35	4.52
74LS74	2	T1	1.58	2.06	2	2.16	4.12	1.32	2.09	2.72	.94	1.49	2.72	.70	1.11	1.44
74S112	1	T1	2.02	16.00	1	2.02	16.00	.66	1.12	9.90	.66	1.12	9.90	.35	.71	5.60
74125	2	T1	1.80	3.49	2	3.60	6.98	1.32	1.11	4.32	.94	1.69	4.32	.70	1.26	2.38
74S253	8	FCILD	3.75	3.75	8	30.00	30.00	5.28	19.80	19.80	3.76	14.10	14.10	2.80	10.50	10.50
74LS258	2	T1	3.94	5.12	2	3.94	10.24	1.32	2.60	7.04	.94	3.70	7.04	.70	2.75	3.58
74S260	2	SIG.	1.32	1.32	2	2.64	2.64	1.32	1.74	1.74	.94	1.24	1.24	.70	.92	.92
74367	2	T1	1.80	1.80	2	3.60	3.60	1.32	1.11	1.11	.94	1.69	1.69	.70	1.26	1.26
74S195	5	T1	6.00	25.11	5	30.00	125.55	3.30	19.80	82.86	2.35	14.10	59.01	1.75	10.50	43.94
25S05	8	AMD	19.50	33.15	8	156.00	265.20	5.28	102.96	175.03	3.76	73.32	124.64	2.80	54.60	92.82
2902	1	T1	5.67	11.40	1	5.67	11.40	.66	3.74	7.52	.66	3.74	7.52	.35	1.98	3.99
74LS175	4	T1	5.25	27.29	4	21.00	109.16	1.88	9.87	51.31	1.88	9.87	51.31	1.40	7.35	38.21
74LS174	8	T1	4.19	22.36	8	33.52	178.88	3.76	15.75	84.07	3.76	15.75	84.07	2.80	32.84	64.00

ITEM	QUANT. PER UNIT	SOURCE	ITEM COST		1 PROCESSOR			500 PROCESSORS			1000 PROCESSORS			10,000 PROCESSORS		
			COMM.	MIL	QUANT. MULT.	COMM.	MIL	QUANT. MULT.	COMM.	MIL	QUANT. MULT.	COMM.	MIL	QUANT. MULT.	COMM.	MIL
74393	2	T1	5.49	5.49	2	10.98	10.98	1.32	7.25	7.25	1.32	7.25	7.25	.70	3.84	3.84
74164	1	T1	4.00	21.82	1	4.00	21.82	.66	2.64	14.40	.66	2.64	14.40	.35	1.40	7.64
74166	1	T1	7.00	21.82	1	7.00	21.82	.66	4.62	14.40	.66	4.62	14.40	.35	2.45	7.64
8T15	1	SIG.	5.20	5.20	1	5.20	5.20	.66	3.43	3.43	.66	3.43	3.43	.35	1.82	1.82
8T16	2	SIG.	7.45	7.45	2	14.90	14.90	1.32	9.83	9.83	.94	7.00	7.00	.70	5.22	5.22
1C PARTS TOTAL			553.33	2055.51		553.33	2055.51		303.54	1013.92		257.01	900.19		216.48	720.77
<u>MEMORY</u>																
93422	8	FCLD	44.90	78.62	8	359.20	628.96	5.28	237.07	415.11	3.76	168.82	295.61	2.80	125.72	220.14
93448	2	FCLD	33.44	56.85	2	66.88	113.70	1.32	44.14	75.04	1.32	44.14	75.04	.94	31.43	53.44
6351	16	MM1	30.00	55.00	16	480.00	880.00	7.52	225.60	413.60	7.52	225.60	413.60	5.60	168.00	308.00
<u>4P CHIPS</u>																
2901	4	AMD	60.00	240.00	4	240.00	960.00	2.64	158.40	633.60	2.64	158.40	633.60	1.88	112.80	451.20
2909	3	AMD	42.12	169.28	3	126.36	507.84	1.98	83.40	335.17	1.98	83.40	335.17	1.41	59.39	238.68
DIGITAL IC's TOTAL			1825.77	5146.01		1825.77	5146.01		1052.14	2886.44		937.37	2653.21		713.82	1992.23
<u>MISC.</u>																
A/D	1		100.00	200.00	1	100.00	200.00	.57	57.00	114.00	.57	57.00	114.00	.32	32.00	64.00
D/A	1		80.	160.	1	80.	160	.57	45.60	91.20	.57	45.60	91.20	.32	25.60	51.20
S/H	1		50.	100.	1	50.	100.	.57	28.50	57.	.57	28.50	57.	.32	16.	32.
Handset	1		40.	80.	1	40.	80.	.57	22.80	45.60	.57	22.80	45.60	.32	12.80	25.60
Lo-Pass	2		50.	100.	2	100.	200.	1.14	57.	114.	1.14	57.	114.	.64	32.	64.
OSC	1		35.	70.	1	35.	70.	.57	19.95	39.90	.57	19.95	39.90	.32	11.20	22.40
Capac.	25		.40	.80	25	10.	20.	.08	3.20	6.40	.08	3.20	6.40	.08	3.20	6.40
Fan	2		25.	50.	2	50.	100.	1.32	33.	66.	.94	23.50	47.	.70	17.50	35.
Power	1		70.	140.	1	70.	140.	.66	46.20	92.40	.66	46.20	92.40	.47	32.90	65.80
PC Board	160 in <sup>2</sup>		160.	160.	1	160.	160.	.66	105.60	105.60	.66	105.60	105.60	.47	75.20	75.20
Package	.2 ft <sup>3</sup>		400	600	1	400.	600.	.57	228.	342.	.57	228.	342.	.32	128.	192.
Connector	4		6.37	18.	4	25.48	72.	2.28	14.52	41.04	2.28	14.52	41.04	1.28	8.15	23.04
Parts Cost			2946.25	6998.01		2946.25	6998.01		1694.76	4001.58		1589.24	3749.35		1108.37	2648.47
HFG. Cost			8838.75	20994.03		8838.75	20994.03		4067.42	9603.79		3496.33	8248.57		2061.57	4105.75

### ACKNOWLEDGEMENTS

The authors gratefully acknowledge the contributions of P.E. Blankenship who did the initial design of the LPCM and Captain R. I. Tucker who wrote the diagnostic system. Many thanks are also due M. L. Malpass whose coding studies of the pitch detector greatly speeded the final design of the LDVT.

## REFERENCES

1. B.S. Atal and S.L. Hanauer, J. Acoust. Soc. Am. 50, pp. 637-655 (1971).
2. J.D. Markel and A.H. Gray, Jr., Linear Prediction of Speech (Springer-Verlag, New York, 1976).
3. E.M. Hofstetter et al., "Vocoder Implementations on the Lincoln Digital Voice Terminal," EASCON '75, Washington, D.C. (29 September-1 October 1975).
4. P.E. Blankenship, "LDVT: High Performance Mini-Computer for Real-Time Speech Processing," EASCON '75, Washington, D.C. (29 September-1 October 1975).
5. P.E. Blankenship, "Preliminary Investigation of Digital Speech Processor Hardware Implementations," Technical Note 1975-8, Lincoln Laboratory, M.I.T. (5 February 1975), DDC AD-A007062/3.
6. J.D. Markel and J.J. Wolf, "Linear Prediction and the Spectral Analysis of Speech," BBN Report No. 2304, Bolt, Beranek and Newman Inc., Cambridge, MA (August 1972).
7. J.D. Markel and A.H. Gray, Jr., IEEE Trans. Acoust., Speech, and Signal Processing ASSP-22, pp. 124-134 (1974).
8. N. Wiener, Extrapolation, Interpolation and Smoothing of Stationary Time Series (The Technology Press and J. Wiley and Sons, New York, 1957), Appendix B.
9. B. Gold and L.R. Rabiner, J. Acoust. Soc. Am. 46, 442 (1969).
10. M.L. Malpass, "The Gold-Rabiner Pitch Detector in a Real-Time Environment," EASCON '75, Washington, D.C. (29 September-1 October 1975).



EXTERNAL DISTRIBUTION

ARPA/IPTO  
1400 Wilson Blvd.  
Arlington, VA 22209  
ATTN: R. Kahn  
W. Marquitz

Defense Communications Agency  
1860 Wiehle Ave.  
Reston, VA 22090  
ATTN: W. Belfield  
G. Coviello  
G. Helm  
R. Sonderegger (10 copies)

Dr. I. L. Lebow  
Defense Communications Agency  
Code 101A  
4135 NDSC 8th and South Courthouse Rd.  
Washington, D.C. 20305

ESD/DCWS  
Mail Stop 22  
Hanscom Field  
Bedford, MA 01730  
ATTN: C.P. Smith  
A.A. Segota  
Lt. S. Meister  
C. Walter

Major Leon Lake  
ESD/DCB  
Mail Stop 49  
Hanscom Field  
Bedford, MA 01731

USC/Information Sciences Institute  
4676 Admiralty Way  
Marina del Rey, CA 90291  
ATTN: D. Cohen  
R. Parker

Steven Boll  
University of Utah  
Computer Science Dept.  
3160 Merrill Engineering Bldg.  
Salt Lake City, Utah 84112

Glen Culler  
Culler-Harrison Inc.  
150-A Aero Camino  
Goleta, CA 93017

D. Forney  
Codex Corporation  
15 River Ave.  
Newton, MA

GTE Sylvania  
Communications System Division  
189 B Street  
Needham, MA  
ATTN: A. Goldberg  
H. Manly

Tom Magill  
Stanford Research Institute  
333 Ravenswood Ave.  
Menlo Park, CA 94025

J. Markel  
SCRL  
800 Miramonte Dr.  
Santa Barbara, CA 93109

John Makhoul  
BBN  
50 Moulton St.  
Cambridge, MA 02138

J. Springer  
Advanced Micro Devices, Inc.  
901 Thompson Place  
Sunnyvale, CA 94086

T. Tremaine  
NSA (R126)  
Fort George G. Meade  
Maryland 20755

Dr. H.S. Edward Tsou  
TRW  
1 Space Park  
Redondo Beach, CA 90278

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-76-302	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle)  Microprocessor Realization of a Linear Predictive Vocoder		5. TYPE OF REPORT & PERIOD COVERED  Technical Note
		6. PERFORMING ORG. REPORT NUMBER Technical Note 1976-37
7. AUTHOR(s)  Edward M. Hofstetter      Omar C. Wheeler Joseph Tierney		8. CONTRACT OR GRANT NUMBER(s)  F19628-76-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M. I. T. P. O. Box 73 Lexington, MA 02173		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Communications Agency 8th Street & So. Courthouse Road Arlington, VA 22204		12. REPORT DATE 30 September 1976
		13. NUMBER OF PAGES 42
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)  Electronic Systems Division Hanscom AFB Bedford, MA 01731		15. SECURITY CLASS. (of this report)  Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES  None		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  microprocessors                                  speech terminals linear predictive vocoder		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  A microprocessor realization for a linear predictive vocoder is presented. The goal was a low power, low cost, compact special purpose realization of a narrow band speech terminal. The resultant design is a general purpose two bus structure running at a 150 ns cycle time using as the basic signal processing element four of the AMD 2901 CPE chips. This basic structure is augmented by a four cycle multiplier to allow for sufficient signal processing power. The design concessions that mark the LPCM as a special purpose machine designed to be a speech terminal are: limited I/O, and limited memory. The present design requires 162 dual-in-line packages, dissipates less than 45 watts and occupies about 1/3 cubic foot.		



